

XML 标记语法

XML 允许在文档中创建、使用新的标记，用户可以根据自己的需要制定适用于自身的标记。标记名称必须遵守如下规则。

- 名称的开头必须是字母、“_”或“.”，但冒号不推荐使用。
- 名称中不能含有空格。
- 名称中还可以包含英文字母、数字、“_”“-”和“.”。

标记的使用规则如下。

- 必须有根标记，且根标记必须唯一。
- 开始标记和结束标记必须配对使用。
- 标记必须合理嵌套，不能交错使用。
- 标记区分大小写。
- 空标记可以写成“<标记名/>”的形式。

XML 文档构成

XML 文档是一种树结构，从“根部”开始，然后扩展到“枝叶”。文档的组成部分包括声明、元素、注释、字符引用和处理指令（在文档中使用显式标记表示），如图-1 所示。

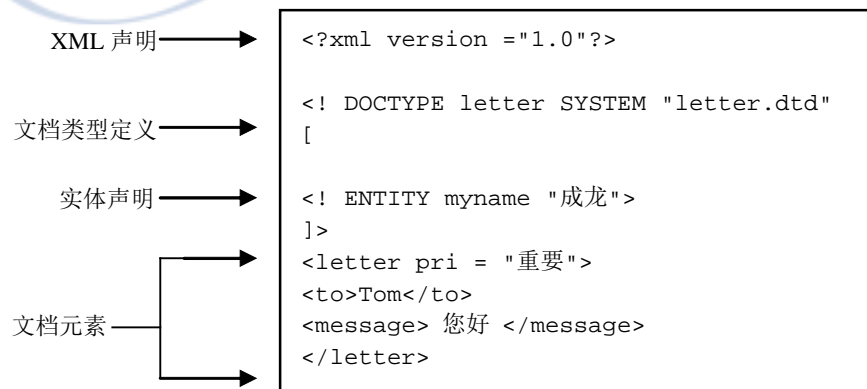


图-1 XML 文档的组成部分

1. XML 声明

XML 声明部分是可选的。

XML 声明的语法：

```
<?xml version="1.0" standalone="no" encoding="UTF-8"?>
```

在语法中：

- version 属性不能省略，用来保证对 XML 版本的支持。
- standalone 属性是可选的，值必须是“yes”或“no”。“yes”表示所有必须的实体声明都包含在文档中，“no”表示需要外部的 DTD。
- encoding 属性是可选的，值必须是一种合法的字符编码。

2. 元素

元素是 XML 文档的基本构成单元，用于表示 XML 文档的结构和数据。元素包含开始标记、数据、结束标记，由于区分大小写，开始标记和结束标记必须完全匹配。

元素的语法：

<标记>包含的内容</标记>

在语法中：

- 每个元素都必须由一个开始标记和一个结束标记分隔开。
- 开始标记是一个包含在尖括号中的元素类型名。
- 结束标记由一个反斜杠和元素类型名组成。
- 如果是不包含任何内容的空元素，可以简写为<标记/>。

3. 属性

属性是附着在元素上，包含有关元素内容的信息，其用于描述元素的某种属性，每个属性都包含一个“名称-值”的组合。

属性的语法：

属性名="属性值"

或

属性名='属性值'

在语法中：

- 属性必须包含在元素的开始标记中，且属性名必须唯一。
- 使用等号分隔属性名和属性值。
- 属性值包含在单引号或双引号中。

4. 字符数据

字符数据是任何不是标记的文本，是元素或属性值的文本内容。字符数据可以是“<”以外的任何合法（Unicode）字符。“<”字符预留作标签的开始字符。字符数据包含以下两类。

- PCDATA：是要通过解析器进行解析的文本，文本中包含的标签将被视为标记，实体将会扩展。
- CDATA：是不通过解析器进行解析的文本，文本中包含的标签将不被视为标记，实体不会扩展。为了便于包含大量的特殊字符，系统提供了 CDATA 块。在 CDATA 块中，XML 解析器会忽略所有的标签和实体引用。

CDATA 的语法：

```
<![CDATA[...]]>
```

在语法中：

- “...”可以是任何字符串，但不能是“]]>”。

5. 注释

用于在文档的 XML 源代码中提供必要的说明。

注释的语法：

```
<!-- 在此处写注释 -->
```

在语法中：

- 注释文本中不应包含“-”或“--”，因为那样可能会使 XML 解析器产生混淆。

注意：注释不能放在标签或实体声明中，不能放在 XML 声明之前。
可以注释标签集，但不能嵌套。

6. 处理指令

处理指令是用来给处理 XML 文档的应用程序提供信息的，XML 分析器把这些信息传递给应用程序，由应用程序来解析这个指令。

处理指令的语法：

<?处理指令名 处理指令信息?>

在语法中：

- 处理指令都必须以“<?”开始，以“?>”结束。

示例：

```
<?xml:stylesheet type= "text/xsl"?>
```

7. 实体

在文档中使用实体可以避免在文档中重复键入长段的文本。我们可以将一个实体名和文本关联，然后每当需要在文档中放入该文本时，就使用实体引用。处理文档时，该实体引用被替换为实体指定的文本。

XML 中包含五个预定义的实体，如表-1 所示。

表-1 预定义实体

实体	用途
&	用来替换字符&
<	用来替换字符小于号(<)
>	用来替换字符大于号(>)
'	用来替换字符串中的单引号(')
"	用来替换字符串中的双引号(")

实体可分为一般实体和参数实体两种。

1) 一般实体

一般实体是可以在 XML 文档中的任何位置出现的实体，包括内部实体和外部实体。

内部实体仅存在于声明它们的文档中。

内部实体的语法：

<!ENTITY ENTITY_NAME "要以实体表示的文本">

内部实体示例：

<!ENTITY address "地址：北京市海淀区成府路 207 号">

外部实体使用一个标识符指向文档外的存储单元。标识符分为两种类型，SYSTEM (系统) 标识符用于引用本地计算机 (或网络)，PUBLIC (公共) 标识符用于引用公共计算机 (或网络)。

外部实体示例：

<!ENTITY greeting SYSTEM "test.txt">

定义实体后，可以在元素内部或属性值中使用实体引用。

实体引用的语法：

&ENTITY_NAME;

在语法中：

- 实体引用以“&”开头，以“;”结束。
- ENTITY_NAME 是已经定义的实体名。

实体引用的规则如下。

- 引用实体前，必须先在 XML 文档中声明该实体。
- 实体引用不应含有任何空格。
- 实体引用的文本必须是格式良好的 XML 文档。
- 引用的文本不应包含“<”字符。

2) 参数实体

当实体和实体引用都只需在 DTD 中出现时，则使用参数实体。

参数实体的语法：

<! ENTITY % PARAMETER_ENTITY_NAME "实体要表示的文本">

参数实体示例：

```
<! ENTITY % ADDRESS "地址：北京市海淀区成府路 207 号">
```

实体引用的语法：

```
%PARAMETER_ENTITY_NAME;
```

8. DOCTYPE 声明

<!DOCTYPE [...]>声明跟在 XML 声明的后面，实体必须在文档 DOCTYPE 中声明。

DOCTYPE 声明的语法：

```
<?xml version="1.0"?>
<!DOCTYPE myDoc>
...在此处声明实体...
<myDoc>
...文档正文...
</myDoc>
```